```
LLL             IIIIIIIII   NNN         NNN KKK         KKK EEEEEEEEEEEEEEE RRRRRRRRRRR
LLL             IIIIIIIII   NNN         NNN KKK         KKK EEEEEEEEEEEEEEE RRRRRRRRRRRR
LLL             IIIIIIIII   NNN         NNN KKK         KKK EEEEEEEEEEEEEEE RRRRRRRRRRRR
LLL                III      NNN         NNN KKK         KKK EEE             RRR       RRR
LLL                III      NNN         NNN KKK         KKK EEE             RRR       RRR
LLL                III      NNN         NNN KKK         KKK EEE             RRR       RRR
LLL                III      NNNNNN      NNN KKK      KKK    EEE             RRR       RRR
LLL                III      NNNNNN      NNN KKK   KKK       EEE             RRR       RRR
LLL                III      NNNNNN      NNN KKKKKKKK        EEE             RRR       RRR
LLL                III      NNN NNN     NNN KKKKKKKK        EEEEEEEEEEE     RRRRRRRRRRRR
LLL                III      NNN  NNN    NNN KKKKKKKK        EEEEEEEEEEE     RRRRRRRRRRRR
LLL                III      NNN   NNN   NNN KKKKKKKK        EEEEEEEEEEE     RRRRRRRRRRRR
LLL                III      NNN    NNNNNN   KKK    KKK      EEE             RRR   RRR
LLL                III      NNN     NNNNNN  KKK      KKK    EEE             RRR    RRR
LLL                III      NNN     NNNNNN  KKK      KKK    EEE             RRR    RRR
LLL                III      NNN         NNN KKK         KKK EEE             RRR       RRR
LLL                III      NNN         NNN KKK         KKK EEE             RRR       RRR
LLL                III      NNN         NNN KKK         KKK EEE             RRR       RRR
LLLLLLLLLLLLLLL    IIIIIIIII NNN        NNN KKK         KKK EEEEEEEEEEEEEEE RRR       RRR
LLLLLLLLLLLLLLL    IIIIIIIII NNN        NNN KKK         KKK EEEEEEEEEEEEEEE RRR       RRR
LLLLLLLLLLLLLLL    IIIIIIIII NNN        NNN KKK         KKK EEEEEEEEEEEEEEE RRR       RRR
```

LNKSYMTBL

LIS

LNK_SYMSERINS
V04=000

D 15
16-Sep-1984 00:36:22   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:37   [LINKER.SRC]LNKSYMTBL.B32;1

Page 1
(1)

```
   1   0001  0  MODULE LNK_SYMSERINS (IDENT='V04-000'
   2   0002  0                        ADDRESSING_MODE(EXTERNAL=GENERAL,
   3   0003  0                                        NONEXTERNAL=LONG_RELATIVE)
   4   0004  0                    ) =
   5   0005  0
   6   0006  1  BEGIN
   7   0007  1
   8   0008  1
   9   0009  1  !***********************************************************************
  10   0010  1  !*                                                                     *
  11   0011  1  !*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
  12   0012  1  !*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
  13   0013  1  !*   ALL RIGHTS RESERVED.                                              *
  14   0014  1  !*                                                                     *
  15   0015  1  !*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
  16   0016  1  !*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
  17   0017  1  !*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
  18   0018  1  !*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
  19   0019  1  !*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
  20   0020  1  !*   TRANSFERRED.                                                       *
  21   0021  1  !*                                                                     *
  22   0022  1  !*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
  23   0023  1  !*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
  24   0024  1  !*   CORPORATION.                                                       *
  25   0025  1  !*                                                                     *
  26   0026  1  !*   DIGITAL ASSUMES NO RESPONSIB'LITY FOR THE USE  OR  RELIABILITY OF ITS *
  27   0027  1  !*   SOFTWARE ON EQUIPMENT WHICH .S NOT SUPPLIED BY DIGITAL.           *
  28   0028  1  !*                                                                     *
  29   0029  1  !*                                                                     *
  30   0030  1  !***********************************************************************
  31   0031  1
  32   0032  1
  33   0033  1
  34   0034  1
  35   0035  1
  36   0036  1  !++
  37   0037  1
  38   0038  1  ! MODULE: LNK_SYMSERINS
  39   0039  1
  40   0040  1  ! FACILITY: LINKER
  41   0041  1
  42   0042  1  ! ABSTRACT: SYMBOL TABLE SEARCH AND INSERT
  43   0043  1
  44   0044  1  ! HISTORY:
  45   0045  1
  46   0046  1  !     VERSION: X01.00
  47   0047  1
  48   0048  1  !     AUTHOR: T.J. PORTER 18-FEB-77
  49   0049  1
  50   0050  1  ! MODIFIED BY:
  51   0051  1
  52   0052  1  !     V03-002 JWT0062        Jim Teague             22-Oct-1982
  53   0053  1  !             Replaced symbol hashing algorithm.
  54   0054  1
  55   0055  1  !     V03-001 JWT0031        Jim Teague             25-May-1982
  56   0056  1  !             Zero hashtable before use for module local symbols
  57   0057  1  !
```

LNK_SYMSERINS
V04=000

E 15
16-Sep-1984 00:36:22   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:37   [LINKER.SRC]LNKSYMTBL.B32;1

Page 2
(1)

```
:   58        0058   1 |
:   59        0059   1 |        V03-003 BLS0090        Benn Schreiber        31-Oct-1981
:   60        0060   1 |                Add LNK$SEARCHLOCAL routine
:   61        0061   1 | |
:   62        0062   1 | |      V03-002 BLS0025        Benn Schreiber        10-Nov-1980
:   63        0063   1 | |              Enhancements to shareable images.
:   64        0064   1 | |
:   65        0065   1 | L      V03-001 BLS0007        Benn Schreiber,        3-Jun-1980
:   66        0066   1 !--              Convert to MDL data structures.
```

```
 68      0067   1  !
 69      0068   1  !++
 70      0069   1  !
 71      0070   1  !     FUNCTIONAL DESCRIPTION:
 72      0071   1  !
 73      0072   1  !          THIS MODULE CONTAINS THE SYMBOL TABLE SEARCH AND INSERT
 74      0073   1  !          ROUTINES. THE SYMBOL TABLE IS A HASH
 75      0074   1  !          TABLE USING A CHAIN OF ALTERNATES WHEN COLLOSIONS OCCUR.
 76      0075   1  !          THE TABLE CONSISTS OF SYMSC TBLSIZ CONTIGUOUS DESCRIPTORS
 77      0076   1  !          EACH DESCRIPTOR CONTAINS A POINTER WHICH IS THE HEAD OF
 78      0077   1  !          A SINGLY LINKED LIST, USED IF A COLLISION OCCURS ON
 79      0078   1  !          THAT ENTRY. MULTIPLE COLLISIONS ARE INSERTED IN THIS LIST,
 80      0079   1  !          THE LAST HAVING A ZERO POINTER.
 81      0080   1  !          THE COLLISION LIST ENTRIES ARE ALLOCATED (BY INSERT) FROM
 82      0081   1  !          DYNAMIC MEMORY.
 83      0082   1  !
 84      0083   1  !          THE CALLING SEQUENCES ARE:
 85      0084   1  !                 LNK$SEARCH (TARGSYMBOL, DESCRADR, SNBADR)
 86      0085   1  !                 LNK$INSERT (TARGSYMBOL, DESCRADR, SNBADR)
 87      0086   1  !                 LNK$SEARCHLOCAL(TARGSYMBOL, ENVINDEX, DESCRADR, SNBADR, ENVDESCADR)
 88      0087   1  !          WHERE:  TARGSYMBOL IS THE ADDRESS OF AN ASCII STRING TO
 89      0088   1  !                       BE FOUND OR INSERTED.
 90      0089   1  !
 91      0090   1  !                  DESCRADR IS THE ADDRESS OF CELL TO STORE THE ADDRESS
 92      0091   1  !                       OF A FOUND ENTRY OR ADDRESS OF AN INSERTED ENTRY.
 93      0092   1  !
 94      0093   1  !                  SNBADR IS THE ADDRESS OF A CELL TO STORE THE ADDRESS
 95      0094   1  !                       OF A FOUND ENTRY SYMBOL NAME BLOCK OR THE ADDRESS
 96      0095   1  !                       OF THE SYMBOL NAME BLOCK IF THE ENTRY WAS INSERTED.
 97      0096   1  !
 98      0097   1  !          SEARCH HAS THE VALUE TRUE IF THE SYMBOL WAS FOUND,
 99      0098   1  !          FALSE IF NOT.
100      0099   1  !
101      0100   1  !          CALLS TO INSERT MUST BE PRECEDED BY AN UNSUCCESSFUL
102      0101   1  !          SEARCH CALL, IN THIS CASE INFORMATION IS
103      0102   1  !          RECORDED TO OBVIATE ANOTHER SEARCH AND
104      0103   1  !          DESCRADR IS RETURNED WITH THE ADDRESS OF THE DESCRIPTOR
105      0104   1  !          WHICH HAS THE STRING COPIED INTO IT.
106      0105   1  !
107      0106   1  !          HASH VALUE H = SUM OF CHARACTERS, DIVIDED BY THE TABLE SIZE.
108      0107   1  !
109      0108   1  !--
110      0109   1  !
111      0110   1  LIBRARY
112      0111   1          'STARLETL32';
113      0112   1  REQUIRE
114      0113   1          'PREFIX';                               ! GET GENERAL DEFINITIONS
115      0228   1  LIBRARY
116      0229   1          'DATBAS';                               ! LINKER DATA STRUCTURES
117      0230   1
118      0231   1  EXTERNAL ROUTINE
119      0232   1          LNK$ALLOBLK,                            ! DYNAMIC MEMORY ALLOCATOR
120      0233   1          LNK$FNDENVMAP;                          ! FIND ENVIRONMENT MAPPING TABLE ENTRY
121      0234   1
122      0235   1  LITERAL
123      0236   1          NINE = 9,
124      0237   1          THIRTEEN = 13;
```

```
; 125        0238 1
; 126        0239 1 EXTERNAL LITERAL
; 127        0240 1         SYM$C_ALLOBLK : BYTLIT;                        ! NUMBER OF PAGES TO PREALLOCATE
; 128        0241 1
; 129        0242 1 OWN
; 130        0243 1         SYMENTRY : REF BLOCK[,BYTE];                   ! CURRENT SYMBOL ENTRY
; 131        0244 1 GLOBAL
; 132        0245 1         LNK$GL_SYMALLOC : VECTOR[2,LONG],              ! SIZE AND POINTER TO PREALLOCTED TABLE
; 133        0246 1         SYM$GL_HASHTBL : REF VECTOR[SYM$C_TBLSIZ,LONG]; ! THE HASH TABLE FOR SYMBOLS
```

LNK_SYMSERINS
V04=000

H 15
16-Sep-1984 00:36:22   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:37   [LINKER.SRC]LNKSYMTBL.B32;1

Page 5
(3)

```
135   0247  1 GLOBAL ROUTINE LNK$SEARCH (TARGSYMBOL, DESCRADR, SNBADR) =
136   0248  1 !
137   0249  2 BEGIN
138   0250  2 !
139   0251  2 !        TARGSYMBOL IS ADDRESS OF AN ASCIC STRING
140   0252  2 !        DESCRADR IS ADDRESS OF CELL TO RECEIVE THE ENTRY
141   0253  2 !        ADDRESS IF SYMBOL IS IN TABLE
142   0254  2 !        SNBADR IS ADDRESS OF CELL TO RECEIVE THE SYMBOL NAME BLOCK
143   0255  2 !        ADDRESS IF SYMBOL IS IN TABLE
144   0256  2 !
145   0257  2 MAP
146   0258  2        TARGSYMBOL : REF VECTOR[,BYTE];
147   0259  2 !
148   0260  2 BUILTIN
149   0261  2        ROT;
150   0262  2
151   0263  2 REGISTER
152   0264  2        HASHINDEX;
153   0265  2
154   0266  2 LOCAL
155   0267  2        LEFTOVER,
156   0268  2        LONGWORDS,
157   0269  2        CH_RESULT,
158   0270  2        PREVENTRY,
159   0271  2        POINTER;
160   0272  2 !
161   0273  2 !        COMPUTE THE HASH INDEX AND GET ENTRY ADDRESS
162   0274  2 !
163   0275  2 HASHINDEX = .TARGSYMBOL[0];                                    ! INITIALIZE THE HASH VALUE
164   0276  2 LEFTOVER = .HASHINDEX AND 3;
165   0277  2 LONGWORDS = .HASHINDEX ^ -2;
166   0278  2 POINTER = TARGSYMBOL[1];                          ! SET CHARACTER POINTER TO INCLUDE STRING LENGTH
167   0279  2 INCR I FROM 1 TO .LONGWORDS DO
168   0280  3        BEGIN
169   0281  3        HASHINDEX = ..POINTER XOR .HASHINDEX;
170   0282  3        POINTER = .POINTER + 4;
171   0283  3        HASHINDEX = ROT (.HASHINDEX ,NINE);
172   0284  2        END;
173   0285  2 INCR I FROM 1 TO  .LEFTOVER DO
174   0286  3        BEGIN
175   0287  3        HASHINDEX = CH$RCHAR_A(POINTER) XOR .HASHINDEX;
176   0288  3        HASHINDEX = ROT ( .HASHINDEX, THIRTEEN);
177   0289  2        END;
178   0290
179   0291  2 HASHINDEX = (.HASHINDEX AND %X'7FFFFFFF') MOD SYM$C_TBLSIZ;          ! THEN TAKE MODULO TABLE SIZE
180   0292  2 SYMENTRY = (SYM$GL_HASHTBL[.HASHINDEX]);                    ! GET ADDRESS OF HASH TABLE ENTRY
181   0293  2 IF .SYMENTRY[SNB$L_COLIST] EQL 0
182   0294  3        THEN RETURN FALSE
183   0295  3        ELSE BEGIN
184   0296  3        PREVENTRY = .SYMENTRY;                          ! REMEMBER PREVIOUS
185   0297  3        SYMENTRY = .SYMENTRY[SNB$L_COLIST];             ! POINT TO THE FIRST ENTRY
186   0298  3 !
187   0299  3 !        NOW COMPARE THE SYMBOL IN THE
188   0300  3 !        ENTRY FOR A MATCH. IF IT MATCHES RETURN ENTRY
189   0301  3 !        ADDRESS AND SUCCESS CONDITION.
190   0302  3 !        IF IT DOES NOT MATCH SEARCH DOWN THE COLLISION
191   0303  3 !        LIST UNTIL:
```

```
  192   0304  3 |           1.      FIND A MATCH - RETURN THE ADDRESS OF MATCHED
  193   0305  3 |                   ENTRY AND SUCCESS
  194   0306  3 |           2.      REACH END OF LIST. SAVE ADDRESS OF LAST
  195   0307  3 |                   ENTRY IN LIST FOR POSSIBLE SUBSEQUENT
  196   0308  3 |                   INSERT AND RETURN FAILURE.
  197   0309  3 |
  198   0310  4 DO IF (CH_RESULT = CH$COMPARE(.TARGSYMBOL[0], TARGSYMBOL[1], ! COMPARE SYMBOLS
  199   0311  3              .SYMENTRY[SNB$B_NAMLNG], SYMENTRY[SNB$T_NAME])) EQL 0
  200   0312  4         THEN BEGIN                                  ! SYMBOL MATCHES
  201   0313  4              .DESCRADR = .SYMENTRY + .SYMENTRY[SNB$B_NAMLNG] + SNB$C_FXDLEN;
  202   0314  4              .SNBADR = .SYMENTRY;                    ! RETURN SYMBOL NAME BLOCK
  203   0315  4                                                     ! AND VALUE BLOCK ADDRESSES
  204   0316  4              RETURN TRUE;                            ! AND RETURN SUCCESS
  205   0317  4         END
  206   0318  4
  207   0319  4 UNTIL (IF .CH_RESULT LSS 0                          ! OTHERWISE, QUIT IF PAST THE SPOT
  208   0320  4              THEN BEGIN
  209   0321  5                   SYMENTRY = .PREVENTRY;       ! RESET POINTER TO INSERT SPOT
  210   0322  5                   TRUE
  211   0323  5              END
  212   0324  5              ELSE BEGIN
  213   0325  5                   PREVENTRY = .SYMENTRY;            ! SAVE PREVIOUS
  214   0326  5                   IF .SYMENTRY[SNB$L_COLIST] EQL 0 ! IF AT END OF LIST
  215   0327  5                        THEN TRUE
  216   0328  6                        ELSE BEGIN
  217   0329  6                             SYMENTRY = .SYMENTRY [SNB$L_COLIST]; ! LINK TO NEXT
  218   0330  6                             FALSE
  219   0331  6                             END
  220   0332  5                   END
  221   0333  3              );
  222   0334  3                                                ! END OF THE COLLISION LIST.
  223   0335  3 RETURN FALSE;                                  ! THE LAST ENTRY EXAMINED
  224   0336  3                                                ! IS PRESERVED IN SYMENTRY.
  225   0337  2         END;                                       ! END OF NON-0 HASH TABLE ENTRY
  226   0338  1 END;                            ! END OF SEARCH ROUTINE


                                 .TITLE   LNK_SYMSERINS
                                 .IDENT   \V04-000\

                                 .PSECT   $OWN$,NOEXE,2

                       00000 SYMENTRY:
                                 .BLKB    4

                                 .PSECT   $GLOBAL$,NOEXE,2

                       00000 LNK$GL_SYMALLOC::
                                 .BLKB    8
                       00008 SYM$GL_HASHTBL::
                                 .BLKB    4

                                 .EXTRN   LNK$ALLOBLK, LNK$FNDENVMAP
                                 .EXTRN   SYM$C_ALLOBLK

                                 .PSECT   $CODE$,NOWRT,2
```

```
                                    07FC 00000              .ENTRY   LNK$SEARCH, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0247
                                                                     R10
                        5A 00000000'  EF  9E 00002          MOVAB    SYMENTRY, R10
                        50          04 BC  9A 00009          MOVZBL   @TARGSYMBOL, HASHINDEX                          0275
        54              50          00 EF  0000D             EXTZV    #0, #2, HASHINDEX, LEFTOVER                     0276
                        53             8F  78 00012          ASHL     #-2, HASHINDEX, LONGWORDS                       0277
                        50          FE AC  D0 00017          MOVL     TARGSYMBOL, R7                                  0278
                        57          04 A7  9E 0001B          MOVAB    1(R7), POINTER
                        52          01 51  D4 0001F          CLRL     I                                              0279
                                       07  11 00021          BRB      2$
                        50             82  CC 00023  1$:      XORL2    (POINTER)+, HASHINDEX                          0281
                        50             09  9C 00026          ROTL     #9, HASHINDEX, HASHINDEX                        0283
                        50             51  F3 0002A  2$:      AOBLEQ   LONGWORDS, I, 1$                               0279
                        F5             51  D4 0002E          CLRL     I                                              0285
                                       0A  11 00030          BRB      4$
                        53             82  9A 00032  3$:      MOVZBL   (POINTER)+, R3                                 0287
                        50             53  CC 00035          XORL2    R3, HASHINDEX
                        50             0D  9C 00038          ROTL     #13, HASHINDEX, HASHINDEX                       0288
                        50             51  F3 0003C  4$:      AOBLEQ   LEFTOVER, I, 3$                                0285
        50              F2          54 1F  00 EF 00040        EXTZV    #0, #31, HASHINDEX, HASHINDEX                  0291
        7E              00          50    01 7A 00045         EMUL     #1, HASHINDEX, #0, -(SP)
        50              50          8E 00000115 8F 7B 0004A   EDIV     #277, (SP)+, HASHINDEX, HASHINDEX
                        6A 00000000'FF40  DE 00053           MOVAL    @SYM$GL_HASHTBL[HASHINDEX], SYMENTRY           0292
                        50             6A  D0 0005B          MOVL     SYMENTRY, R0                                   0293
                                       60  D5 0005E          TSTL     (R0)
                                       47  13 00060          BEQL     9$
                        58             50  D0 00062          MOVL     R0, PREVENTRY                                  0296
                        6A             60  D0 00065          MOVL     (R0), SYMENTRY                                 0297
                        50          04 BC  9A 00068  5$:      MOVZBL   @TARGSYMBOL, R0                                0310
                        54             6A  D0 0006C          MOVL     SYMENTRY, R4                                   0311
                        55          04 A4  9A 0006F          MOVZBL   4(R4), R5
                        56             01  D0 00073          MOVL     #1, R6
        55              00          01 A7  50 2D 00076        CMPC5    R0, 1(R7), #0, R5, 5(R4)
                                       05  A4 0007C
                                       03  1A 0007E          BGTRU    6$
                        56             01  D9 00080          SBWC     #1, R6
                        59             56  D0 00083  6$:      MOVL     R6, CH_RESULT
                                       0E  12 00086          BNEQ     7$
                        08 BC       05 A544 9E 00088          MOVAB    5(R5)[R4], @DESCRADR                           0313
                        0C BC          54  D0 0008E          MOVL     R4, @SNBADR                                    0314
                        50             01  D0 00092          MOVL     #1, R0                                         0316
                                       04  00096            RET
                                       05  18 00096  7$:      BGEQ     8$                                            0319
                        6A             58  D0 00098          MOVL     PREVENTRY, SYMENTRY                            0321
                                       0C  11 0009B          BRB      9$
                        58             54  D0 0009D  8$:      MOVL     R4, PREVENTRY                                 0325
                                       64  D5 000A0          TSTL     (R4)                                          0326
                                       05  13 000A2          BEQL     9$
                        6A             64  D0 000A4          MOVL     (R4), SYMENTRY                                 0329
                                       BF  11 000A7          BRB      5$
                        50             D4 000A9  9$:          CLRL     R0                                           0335
                                       04 000AB            RET                                                      0338
```

```
; Routine Size:  172 bytes,    Routine Base:  $CODE$ + 0000
```

LNK_SYMSERINS
V04=000

K 15
16-Sep-1984 00:36:22   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:37   [LINKER.SRC]LNKSYMTBL.B32;1

Page 8
      (4)

```
228  0339  1  GLOBAL ROUTINE LNK$INSERT(TARGSYMBOL, DESCRADR, SNBADR) : NOVALUE =
229  0340  2  BEGIN
230  0341  2
231  0342  2  !       TARGSYMBOL IS ADDRESS OF AN ASCIC STRING, AN ENTRY
232  0343  2  !       FOR WHICH IS TO BE INSERTED IN THE SYMBOL TABLE. THE
233  0344  2  !       ADDRESS OF THIS ENTRY IS TO BE RETURNED IN THE CELL
234  0345  2  !       DESCRADR.  THE ADDRESS OF THE SYMBOL NAME BLOCK IS RETURNED
235  0346  2  !       IN THE CELL POINTED TO BY SNBADR.  THIS ROUTINE REQUIRES THAT
236  0347  2  !       AN UNSUCCESSFULL CALL ON SEARCH PRECEDED IT AND SAVED THE
237  0348  2  !       ADDRESS OF THE LAST ENTRY EXAMINED.
238  0349  2  !
239  0350  2  MAP
240  0351  2          TARGSYMBOL : REF VECTOR[, BYTE];
241  0352  2  LOCAL
242  0353  2          BLOCKSIZE,
243  0354  2          NEWENTRY : REF BLOCK[,BYTE];
244  0355  2  !
245  0356  2  BLOCKSIZE = (SYM$C_SIZE+SNB$C_FXDLEN+.TARGSYMBOL[0] + 3) AND NOT 3;
246  0357  2  IF .LNK$GL_SYMALLOC[0] LEQU .BLOCKSIZE
247  0358  3  THEN BEGIN
248  0359  3      LNK$ALLOBLK(SYM$C_ALLOBLK+512,LNK$GL_SYMALLOC[1]);
249  0360  3      LNK$GL_SYMALLOC[0] = SYM$C_ALLOBLK+512;
250  0361  2      END;
251  0362  2  NEWENTRY = .LNK$GL_SYMALLOC[1];
252  0363  2  LNK$GL_SYMALLOC[0] = .LNK$GL_SYMALLOC[0] - .BLOCKSIZE;          ! ALLOCATE A BLOCK
253  0364  2  LNK$GL_SYMALLOC[1] = .LNK$GL_SYMALLOC[1] + .BLOCKSIZE;
254  0365  2                                                                 ! WHICH CONSISTS OF
255  0366  2                                                                 ! SYMBOL VALUE BLOCK +
256  0367  2                                                                 ! SIZE OF NAME
257  0368  2                                                                 ! + NAME BLOCK OVERHEAD
258  0369  2  NEWENTRY[SNB$L_COLIST] = .SYMENTRY[SNB$L_COLIST];              ! LINK INTO THE LIST
259  0370  2  SYMENTRY[SNB$L_COLIST] = .NEWENTRY;                            ! LINK IT ON TO COLLISION LIST
260  0371  2  SYMENTRY = .NEWENTRY;                                          ! AND REPLACE OLD POINTER
261  0372  2
262  0373  2  !       HAVE AN EMPTY DESCRIPTOR - COPY IN THE SYMBOL
263  0374  2  !       STRING
264  0375  2  !
265  0376  2  CH$MOVE(.TARGSYMBOL[0]+1,TARGSYMBOL[0],                        ! COPY NAME
266  0377  2              SYMENTRY[SNB$B_NAMLNG]);                           ! (NO EXTRA BYTES IN NAME)
267  0378  2  .SNBADR = .SYMENTRY;                                           ! RETURN SYMBOL NAME BLOCK ADDRESS
268  0379  2  SYMENTRY = .SYMENTRY + .TARGSYMBOL[0] + SNB$C_FXDLEN;          ! POINT TO SYMBOL VALUE BLOCK
269  0380  2  CH$FILL(0,SYM$C_SIZE,,SYMENTRY);                               ! ZERO THE ENTRY
270  0381  2  SYMENTRY[SYM$B_NAMLNG] = .TARGSYMBOL[0];                       ! SET LENGTH INTO VALUE BLOCK
271  0382  2  .DESCRADR = .SYMENTRY;                                         ! RETURN ITS ADDRESS
272  0383  2  RETURN;                                                        ! AND THAT'S IT
273  0384  1  END;                                                          ! OF INSERT ROUTINE.


                              03FC 00000         .ENTRY  LNK$INSERT, Save R2,R3,R4,R5,R6,R7,R8,R9  ; 0339
              59 00000000'  EF  9E 00002         MOVAB   LNK$GL_SYMALLOC, R9
              58 00000000'  EF  9E 00009         MOVAB   SYMENTRY, R8
              57        04  BC  9A 00010         MOVZBL  @TARGSYMBOL, R7
              50        2C  A7  9E 00014         MOVAB   44(R7), R0
        52    50        03  CB 00018             BICL3   #3, R0, BLOCKSIZE                         ; 0356
```

```
                        52            69 D1 0001C        CMPL    LNK$GL_SYMALLOC, BLOCKSIZE        : 0357
                                      17 1A 0001F        BGTRU   1$
                              04      A9 9F 00021        PUSHAB  LNK$GL_SYMALLOC+4                 : 0359
                  00000000* 8F DD 00024                 PUSHL   #<SYM$C_ALLOBLK*512>
        00000000G 00                  02 FB 0002A        CALLS   #2, LNK$ALLOBLK
                  69 00000000* 8F D0 00031              MOVL    #<SYM$C_ALLOBLK*512>, LNK$GL_SYMALLOC   : 0360
                  50           04 A9 D0 00038 1$:        MOVL    LNK$GL_SYMALLOC+4, NEWENTRY      : 0362
                  69              52 C2 0003C            SUBL2   BLOCKSIZE, LNK$GL_SYMALLOC       : 0363
              04  A9              52 C0 0003F            ADDL2   BLOCKSIZE, LNK$GL_SYMALLOC+4     : 0364
                  60        00    B8 D0 00043            MOVL    @SYMENTRY, (NEWENTRY)            : 0369
                  00        B8    50 D0 00047            MOVL    NEWENTRY, @SYMENTRY             : 0370
                  68              50 D0 0004B            MOVL    NEWENTRY, SYMENTRY             : 0371
                  50           01 A7 9E 0004E            MOVAB   1(R7), R0                      : 0376
                  56              68 D0 00052            MOVL    SYMENTRY, R6                   : 0377
          04 A6   04 BC           50 28 00055           MOVC3   R0, @TARG$YMBOL, 4(R6)        : 0378
                  0C BC           56 D0 0005B           MOVL    R6, @SNBADR                   : 0379
                  68      05 A746 9E 0005F              MOVAB   5(R7)[R6], SYMENTRY          : 0380
                  56              68 D0 00064           MOVL    SYMENTRY, R6
          24      00      6E      00 2C 00067           MOVC5   #0, (SP), #0, #36, (R6)
                                  66    0006C
                  0F A6           57 90 0006D           MOVB    R7, 15(R6)                   : 0381
                  08 BC           56 D0 00071           MOVL    R6, @DESCRADR                : 0382
                                  04    00075           RET                                  : 0384
```

; Routine Size: 118 bytes,   Routine Base: $CODE$ + 00AC

LNK_SYMSERINS
V04=000

M 15
16-Sep-1984 00:36:22   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:37   [LINKER.SRC]LNKSYMTBL.B32;1

Page 10
(5)

```
275    0385    1  GLOBAL ROUTINE LNK$SEARCHLOCAL (TARGSYMBOL, ENVINDEX, DESCRADR, SNBADR, ENVDESCADR) =
276    0386    1  !
277    0387    2  BEGIN
278    0388    2  !
279    0389    2  !    TARGSYMBOL IS ADDRESS OF AN ASCIC STRING
280    0390    2  !    ENVINDEX IS THE ENVIRONMENT THAT SYMBOL IS FROM
281    0391    2  !    DESCRADR IS ADDRESS OF CELL TO RECEIVE THE ENTRY
282    0392    2  !         ADDRESS IF SYMBOL IS IN TABLE
283    0393    2  !    SNBADR IS ADDRESS OF CELL TO RECEIVE THE SYMBOL NAME BLOCK
284    0394    2  !         ADDRESS IF SYMBOL IS IN TABLE
285    0395    2  !    ENVDESCADR IS ADDRESS OF CELL TO RECEIVE THE ENVIRONMENT
286    0396    2  !         DESCRIPTOR BLOCK ADDRESS OR 0 IF NOT DEFINED
287    0397    2  !         OR REFERENCED (OPTIONAL PARAMETER)
288    0398    2  !
289    0399    2  MAP
290    0400    2       TARGSYMBOL : REF VECTOR[,BYTE];
291    0401    2
292    0402    2  BUILTIN
293    0403    2       NULLPARAMETER;
294    0404    2
295    0405    2  BUILTIN
296    0406    2       ROT;
297    0407    2
298    0408    2  REGISTER
299    0409    2       HASHINDEX;
300    0410    2
301    0411    2  LOCAL
302    0412    2       LONGWORDS,
303    0413    2       LEFTOVER,
304    0414    2       MAPENT : REF BLOCK[,BYTE],
305    0415    2       ENVDESC : REF BLOCK[,BYTE],
306    0416    2       ENVNODE : REF BLOCK[,BYTE],
307    0417    2       HASHTABLE : REF VECTOR[,LONG],
308    0418    2       CH_RESULT,
309    0419    2       PREVENTRY,
310    0420    2       POINTER;
311    0421    2  !
312    0422    2  !    COMPUTE THE HASH INDEX
313    0423    2  !
314    0424    2  HASHINDEX = .TARGSYMBOL[0];                                      ! INITIALIZE THE HASH VALUE
315    0425    2  LEFTOVER = .HASHINDEX AND 3;
316    0426    2  LONGWORDS = .HASHINDEX ^ -2;
317    0427    2  POINTER = TARGSYMBOL[1];                                         ! SET CHARACTER POINTER TO INCLUDE STRING LENGTH
318    0428    2  INCR I FROM 1 TO .LONGWORDS DO
319    0429    2       BEGIN
320    0430    2       HASHINDEX = ..POINTER XOR .HASHINDEX;
321    0431    2       POINTER = .POINTER + 4;
322    0432    2       HASHINDEX = ROT (.HASHINDEX ,NINE);
323    0433    2       END;
324    0434    2  INCR I FROM 1 TO  .LEFTOVER DO
325    0435    2       BEGIN
326    0436    2       HASHINDEX = CH$RCHAR_A(POINTER) XOR .HASHINDEX;
327    0437    2       HASHINDEX = ROT ( .HASHINDEX, THIRTEEN);
328    0438    2       END;
329    0439    2
330    0440    2  HASHINDEX = (.HASHINDEX AND %X'7FFFFFFF') MOD SYM$C_TBLSIZ;      ! THEN TAKE MODULO TABLE SIZE
331    0441    2  !
```

LNK_SYMSERINS
V04=000

N 15
16-Sep-1984 00:36:22   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:37   [LINKER.SRC]LNKSYMTBL.B32;1

Page 11
(5)

```
332    0442  2 ! FIND ENVIRONMENT SYMBOL HASH TABLE
333    0443  2 !
334    0444  2 MAPENT = LNK$FNDENVMAP(.ENVINDEX);
335    0445  2 IF (ENVNODE = .MAPENT[PMT$L_PSCDES]) NEQ 0
336    0446  2 THEN BEGIN
337    0447          ENVDESC = .ENVNODE + NODE$C_SHORT;
338    0448          HASHTABLE = .ENVDESC[NVD$L_SYMTBL];
339    0449          END
340    0450  2 ELSE BEGIN
341    0451          HASHTABLE = .MAPENT[PMT$L_SYMLST];
342    0452          ENVDESC = 0;
343    0453          END;
344    0454  2 IF .HASHTABLE EQL 0
345    0455  2 THEN BEGIN
346    0456          LNK$ALLOBLK(SYM$C_TBLSIZ*4,HASHTABLE);        ! ALLOCATE TABLE AND
347    0457          CH$FILL(0,SYM$C_TBLSIZ*4,.HASHTABLE);         ! ZERO IT BEFORE USE
348    0458          IF .ENVNODE NEQ 0
349    0459              THEN ENVDESC[NVD$L_SYMTBL] = .HASHTABLE
350    0460              ELSE MAPENT[PMT$L_SYMLST] = .HASHTABLE;
351    0461          END;
352    0462  2 IF NOT NULLPARAMETER(5)
353    0463          THEN .ENVDESCADR = .ENVDESC;
354    0464  2 SYMENTRY = (HASHTABLE[.HASHINDEX]);                  ! GET ADDRESS OF HASH TABLE ENTRY
355    0465  2 IF .SYMENTRY[SNB$L_COLIST] EQL 0
356    0466          THEN RETURN FALSE
357    0467          ELSE BEGIN
358    0468          PREVENTRY = .SYMENTRY;                        ! REMEMBER PREVIOUS
359    0469          SYMENTRY = .SYMENTRY[SNB$L_COLIST];           ! POINT TO THE FIRST ENTRY
360    0470  !
361    0471  !    NOW COMPARE THE SYMBOL IN THE
362    0472  !    ENTRY FOR A MATCH. IF IT MATCHES RETURN ENTRY
363    0473  !    ADDRESS AND SUCCESS CONDITION.
364    0474  !    IF IT DOES NOT MATCH SEARCH DOWN THE COLLISION
365    0475  !    LIST UNTIL:
366    0476  !    1.      FIND A MATCH - RETURN THE ADDRESS OF MATCHED
367    0477  !            ENTRY AND SUCCESS
368    0478  !    2.      REACH END OF LIST. SAVE ADDRESS OF LAST
369    0479  !            ENTRY IN LIST FOR POSSIBLE SUBSEQUENT
370    0480  !            INSERT AND RETURN FAILURE.
371    0481  !
372    0482  4 DO IF (CH_RESULT = CH$COMPARE(.TARGSYMBOL[0], TARGSYMBOL[1], ! COMPARE SYMBOLS
373    0483  3                    .SYMENTRY[SNB$B_NAMLNG], SYMENTRY[SNB$T_NAME])) EQL 0
374    0484  4          THEN BEGIN                                   ! SYMBOL MATCHES
375    0485  4              .DESCRADR = .SYMENTRY + .SYMENTRY[SNB$B_NAMLNG] + SNB$C_FXDLEN;
376    0486  4              .SNBADR = .SYMENTRY;                     ! RETURN SYMBOL NAME BLOCK
377    0487  4                                                      ! AND VALUE BLOCK ADDRESSES
378    0488  4              RETURN TRUE;                             ! AND RETURN SUCCESS
379    0489  4              END
380    0490  4
381    0491  4 UNTIL (IF .CH_RESULT LSS 0                            ! OTHERWISE, QUIT IF PAST THE SPOT
382    0492                    THEN BEGIN
383    0493  5                      SYMENTRY = .PREVENTRY;           ! RESET POINTER TO INSERT SPOT
384    0494  5                      TRUE
385    0495  5                      END
386    0496                      ELSE BEGIN
387    0497  5                      PREVENTRY = .SYMENTRY;           ! SAVE PREVIOUS
388    0498  5                      IF .SYMENTRY[SNB$L_COLIST] EQL 0 ! IF AT END OF LIST
```

```
389    0499  5                          THEN TRUE
390    0500  6                          ELSE BEGIN
391    0501  6                              SYMENTRY = .SYMENTRY [SNBSL_COLIST]; ! LINK TO NEXT
392    0502  6                              FALSE
393    0503  6                              END
394    0504  5                      END
395    0505  5            );                END
396    0506  3                                   ! END OF THE COLLISION LIST.
397    0507  3            RETURN FALSE;          ! THE LAST ENTRY EXAMINED
398    0508  3                                   ! IS PRESERVED IN SYMENTRY.
399    0509  2            END;                        ! END OF NON-0 HASH TABLE ENTRY
400    0510  1  END;                       ! END OF SEARCH ROUTINE
```

```
                                       OFFC 00000      .ENTRY  LNK$SEARCHLOCAL, Save R2,R3,R4,R5,R6,R7,R8,-;  0385
                                                               R9,R10,R11
                         5B 00000000' EF  9E 00002      MOVAB   SYMENTRY, R11
                 53                56    04 BC  9A 00009 MOVZBL @TARGSYMBOL, HASHINDEX                         0424
                                    02    00 EF 0000D    EXTZV   #0, #2, HASHINDEX, LEFTOVER                   0425
                                    52    FE 8F  78 00012 ASHL   #-2, HASHINDEX, LONGWORDS                     0426
                                    5A    04 AC  D0 00017 MOVL   TARGSYMBOL, R10                               0427
                                    51    01 AA  9E 0001B MOVAB  1(R10), POINTER
                                    50       D4 0001F    CLRL    I                                             0428
                                    07       10 00021    BSBB    2$
                 56                56    81 CC 00023 1$:  XORL2  (POINTER)+, HASHINDEX                          0430
                                    56    09 9C 00026     ROTL   #9, HASHINDEX, HASHINDEX                      0432
                 F5                50    52 F3 0002A 2$:  AOBLEQ LONGWORDS, I, 1$                               0428
                                    50       D4 0002E    CLRL    I                                             0434
                                    0A       11 00030    BRB     4$
                                    52    81 9A 00032 3$: MOVZBL (POINTER)+, R2                                0436
                                    56    52 CC 00035     XORL2  R2, HASHINDEX                                 0437
                 56                56    0D 9C 00038     ROTL   #13, HASHINDEX, HASHINDEX                      0437
                 F2                50    53 F3 0003C 4$:  AOBLEQ LEFTOVER, I, 3$                                0434
                 56                56    00 EF 00040     EXTZV   #0, #31, HASHINDEX, HASHINDEX                  0440
                 7E                00    56 7A 00045     EMUL    #1, HASHINDEX, #0, -(SP)
                 56                56    8E 00000115 8F 7B 0004A EDIV #277, (SP)+, HASHINDEX, HASHINDEX
                                    08    AC DD 00053    PUSHL   ENVINDEX                                      0444
                       00000000G   00    01 FB 00056    CALLS   #1, LNK$FNDENVMAP
                                    59    50 D0 0005D    MOVL    R0, MAPENT
                                    58    69 D0 00060    MOVL    (MAPENT), ENVNODE                             0445
                                    0A       13 00063    BEQL    5$
                 57       0A       A8    57 9E 00065    MOVAB   10(R8), ENVDESC                               0447
                 6E       08       A7    6E D0 00069    MOVL    8(ENVDESC), HASHTABLE                          0448
                                    06       11 0006D    BRB     6$                                            0445
                 6E       04       A9    6E D0 0006F 5$: MOVL   4(MAPENT), HASHTABLE                           0451
                                    57       D4 00073    CLRL    ENVDESC                                       0452
                                    6E       D5 00075 6$: TSTL   HASHTABLE                                     0454
                                    25       12 00077    BNEQ    8$
                                    5E       DD 00079    PUSHL   SP                                            0456
                 7E       0454      8F    5E 3C 0007B    MOVZWL  #1108, -(SP)
                       00000000G   00    02 FB 00080    CALLS   #2, LNK$ALLOBLK
      0454   BF              00    6E    00 2C 00087    MOVC5   #0, (SP), #0, #1108, @HASHTABLE               0457
                                    00    BE    0008E
                                    58    D5 00090    TSTL    ENVNODE                                         0458
```

```
                              06  13 00092          BEQL    7$
                   0B  A7     6E  D0 00094          MOVL    HASHTABLE, 8(ENVDESC)        0459
                              04  11 00098          BRB     8$
                   04  A9     6E  D0 0009A 7$:      MOVL    HASHTABLE, 4(MAPENT)         0460
                       05     6C  91 0009E 8$:      CMPB    (AP), #5                     0462
                              09  1F 000A1          BLSSU   9$
                          14  AC  D5 000A3          TSTL    20(AP)
                              04  15 000A6          BEQL    9$
                   14  BC     57  D0 000A8          MOVL    ENVDESC, @ENVDESCADR         0463
                       6B 00 BE46 DE 000AC 9$:      MOVAL   @HASHTABLE[HASHINDEX], SYMENTRY  0464
                       50     6B  D0 000B1          MOVL    SYMENTRY, R0                 0465
                              60  D5 000B4          TSTL    (R0)
                              47  15 000B6          BEQL    14$
                       56     50  D0 000B8          MOVL    R0, PREVENTRY                0468
                       6B     60  D0 000BB          MOVL    (R0), SYMENTRY               0469
                       50  04 BC  9A 000BE 10$:     MOVZBL  @TARGSYMBOL, R0              0482
                       54     6B  D0 000C2          MOVL    SYMENTRY, R4                 0483
                       55  04 A4  9A 000C5          MOVZBL  4(R4), R5
                       57     01  D0 000C9          MOVL    #1, R7
    55        00     01  AA     50  2D 000CC        CMPC5   R0, 1(R10), #0, R5, 5(R4)
                          05  A4     000D2
                              03  1A 000D4          BGTRU   11$
                       57     01  D9 000D6          SBWC    #1, R7
                       58     57  D0 000D9 11$:     MOVL    R7, CH_RESULT
                              0E  12 000DC          BNEQ    12$
                   0C  BC  05 A544 9E 000DE         MOVAB   5(R5)[R4], @DESCRADR         0485
                   10  BC     54  D0 000E4          MOVL    R4, @SNBADR                  0486
                       50     01  D0 000E8          MOVL    #1, R0                       0488
                              04 000EB             RET
                              05  18 000EC 12$:     BGEQ    13$                          0491
                       SB     56  D0 000EE          MOVL    PREVENTRY, SYMENTRY          0493
                              0C  11 000F1          BRB     14$
                       56     54  D0 000F3 13$:     MOVL    R4, PREVENTRY                0497
                              64  D5 000F6          TSTL    (R4)                         0498
                              05  13 000F8          BEQL    14$
                       6B     64  D0 000FA          MOVL    (R4), SYMENTRY               0501
                              BF  11 000FD          BRB     10$
                       50     D4 000FF 14$:         CLRL    R0                           0507
                              04 00101             RET                                   0510
```

; Routine Size: 25B bytes,   Routine Base: $CODE$ + 0122

```
; 402    0511  1 GLOBAL ROUTINE LNK$UPCASE_D (DESCR) =
; 403    0512  2 BEGIN
; 404    0513  2 !
; 405    0514  2 ! THIS ROUTINE UPCASES THE STRING DESCRIBED BY DESCR.
; 406    0515  2 !
; 407    0516  2 MAP
; 408    0517  2     DESCR : REF BBLOCK;
; 409    0518  2
; 410    0519  2 BIND
; 411    0520  2     BYTESTRING = .DESCR[DSC$A_POINTER] : VECTOR[,BYTE];
; 412    0521  2
; 413    0522  2 LOCAL
; 414    0523  2     CCHAR : BYTE;
; 415    0524  2
; 416    0525  2 IF .DESCR[DSC$W_LENGTH] NEQ 0
; 417    0526  2 THEN INCRU I FROM 0 TO .DESCR[DSC$W_LENGTH] - 1
; 418    0527  2     DO IF (CCHAR = .BYTESTRING[.I]) GEQU %ASCII 'a'
; 419    0528  2         AND .CCHAR LEQU %ASCII 'z'
; 420    0529  2     THEN BYTESTRING[.I] = .CCHAR - 32;
; 421    0530  2
; 422    0531  2 RETURN TRUE
; 423    0532  1 END;
```

```
                           001C 00000        .ENTRY  LNK$UPCASE_D, Save R2,R3,R4      ; 0511
              50     04 AC D0 00002           MOVL    DESCR, R0                       ; 0520
                        60 B5 00006           TSTW    (R0)                            ; 0525
                        2A 13 00008           BEQL    4$
              54        60 3C 0000A           MOVZWL  (R0), R4                        ; 0526
                        54 D7 0000D           DECL    R4
                        51 D4 0000F           CLRL    I
                        1C 11 00011           BRB     3$
                 52  04 B041 9A 00013 1$:     MOVZBL  a4(R0)[I], R2                   ; 0527
                 53        52 90 00018        MOVB    R2, CCHAR
           61 8F          52 91 0001B        CMPB    R2, #97
                        0C 1F 0001F           BLSSU   2$
           7A 8F          53 91 00021        CMPB    CCHAR, #122                      ; 0528
                        06 1A 00025           BGTRU   2$
     04 B041             53 20 83 00027       SUBB3   #32, CCHAR, a4(R0)[I]          ; 0529
                        51 D6 0002D 2$:       INCL    I                              ; 0527
              54        51 D1 0002F 3$:       CMPL    I, R4
                        DF 1B 00032           BLEQU   1$
              50        01 D0 00034 4$:       MOVL    #1, R0                         ; 0531
                        04 00037              RET                                     ; 0532
```

; Routine Size: 56 bytes,   Routine Base: $CODE$ + 0224

```
; 424    0533  1 GLOBAL ROUTINE LNK$UPCASE_C (STRINGADR) =
; 425    0534  2 BEGIN
; 426    0535  2 !
; 427    0536  2 ! THIS ROUTINE UPCASES THE ASCIC STRING POINTED TO BY STRINGADR
; 428    0537  2 !
; 429    0538  2 MAP
```

```
;  430        0539  2            STRINGADR : REF VECTOR[,BYTE];
;  431        0540  2
;  432        0541  2    LOCAL
;  433        0542  2            CCHAR : BYTE;
;  434        0543  2
;  435        0544  2    IF .STRINGADR[0] NEQ 0
;  436        0545  2    THEN INCRU I FROM 1 TO .STRINGADR[0]
;  437        0546  2        DO IF (CCHAR = .STRINGADR[.I]) GEQU %ASCII 'a'
;  438        0547  2            AND .CCHAR LEQU %ASCII 'z'
;  439        0548  2            THEN STRINGADR[.I] = .CCHAR - 32;
;  440        0549  2
;  441        0550  2    RETURN TRUE
;  442        0551  1    END;
```

```
                                    000C 00000          .ENTRY   LNK$UPCASE_C, Save R2,R3        ; 0533
                        52      04  BC  9A 00002         MOVZBL   @STRINGADR, R2                  ; 0544
                                    26  13 00006         BEQL     4$
                        50          01  D0 00008         MOVL     #1, I                           ; 0545
                                    1C  11 0000B         BRB      3$
                        51      04 BC40 9A 0000D 1$:     MOVZBL   @STRINGADR[I], R1               ; 0546
                        53          51  90 00012         MOVB     R1, CCHAR
              61    8F  51          51  91 00015         CMPB     R1, #97
                                    0C  1F 00019         BLSSU    2$
              7A    8F  53          53  91 0001B         CMPB     CCHAR, #122                     ; 0547
                                    06  1A 0001F         BGTRU    2$
      04 BC40         53            20  83 00021         SUBB3    #32, CCHAR, @STRINGADR[I]       ; 0548
                                    50  D6 00027 2$:     INCL     I                               ; 0546
                        52          50  D1 00029 3$:     CMPL     I, R2
                                    DF  1B 0002C         BLEQU    1$
                        50          01  D0 0002E 4$:     MOVL     #1, R0                          ; 0550
                                    04 00031            RET                                       ; 0551
```

; Routine Size: 50 bytes,   Routine Base: $CODE$ + 025C

; 443   0552 0 END ELUDOM

                            PSECT SUMMARY

;   Name   Bytes   Attributes

; OWN 4 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
; $GLOBAL$ 12 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
; $CODE$ 654 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

```
;                        Library Statistics
;
;                                  -------- Symbols --------   Pages     Processing
;          File                    Total  Loaded   Percent    Mapped    Time
;
;    _$255$DUA28:[SYSLIB]STARLET.L32;1    9776      6        0        581       00:01.0
;    _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1  538     11        2         28       00:00.4




:                          COMMAND QUALIFIERS

;        BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LNKSYMTBL/OBJ=OBJ$:LNKSYMTBL MSRC$:LNKSYMTBL/UPDATE=(ENH$:LNKSYMTBL)

; Size:          654 code + 16 data bytes
; Run Time:         00:14.6
; Elapsed Time:     00:44.9
; Lines/CPU Min:    2266
; Lexemes/CPU-Min: 14751
; Memory Used:  111 pages
; Compilation Complete
```